**SynSys Documentation**

**Summary**

SynSys is a system written in Python 3.6 used to generate synthetic sensor event data using a real smart home data file as a basis. SynSys uses Hidden Markov Models (HMMs) to generate activity and sensor event sequences and regression to generate timestamps.

**Python Module Dependencies**

1. Cython: <http://cython.org/>
2. Scipy: https://www.scipy.org/
3. Numpy: http://www.numpy.org/
4. Pandas: <https://pandas.pydata.org/>
5. Setup tools (this may already be installed on your system): <https://setuptools.readthedocs.io/en/latest/>
6. Sklearn: <http://scikit-learn.org/stable/index.html>
7. Statsmodels: http://www.statsmodels.org/devel/generated/statsmodels.discrete.discrete\_model.Poisson.fit.html

SynSys also uses a Python module called pomegranate for the implementation of the Hidden Markov Models written by Jacob Schreiber. This code is included in the distribution of SynSys. Pomegranate is distributed under an MIT license. You can you replace the pomegranate code with another implementation of Hidden Markov Models if desired.

If you are using Windows there may be some extra steps involved to get Cython working that are not needed for Linux or macOS:

1. Install Python 3.6 for Windows (x64) from Python website (including Python in system path).
2. pip install Cython
3. pip install numpy scipy
4. pip install pandas
5. pip install scikit-learn
6. I had to install the Visual C++ Build Tools Standalone from here in order for Cython to be able to compile the package: <https://wiki.python.org/moin/WindowsCompilers#Microsoft_Visual_C.2B-.2B-_14.0_standalone:_Build_Tools_for_Visual_Studio_2017_.28x86.2C_x64.2C_ARM.2C_ARM64.29>
7. Run the setup command from the documentation:  
   python setup.py build\_ext --inplace install
8. Run the program from the main directory:  
   python synsys/runSynSys.py ./hh111week1dataclusters.csv 10 10 6 1  
   The command outputs sensor events after some time.

**How to install**

After installing dependencies, run:

python setup.py build\_ext –inplace install

**How to run**

Once you have run setup.py you can use SynSys via the command line like this:

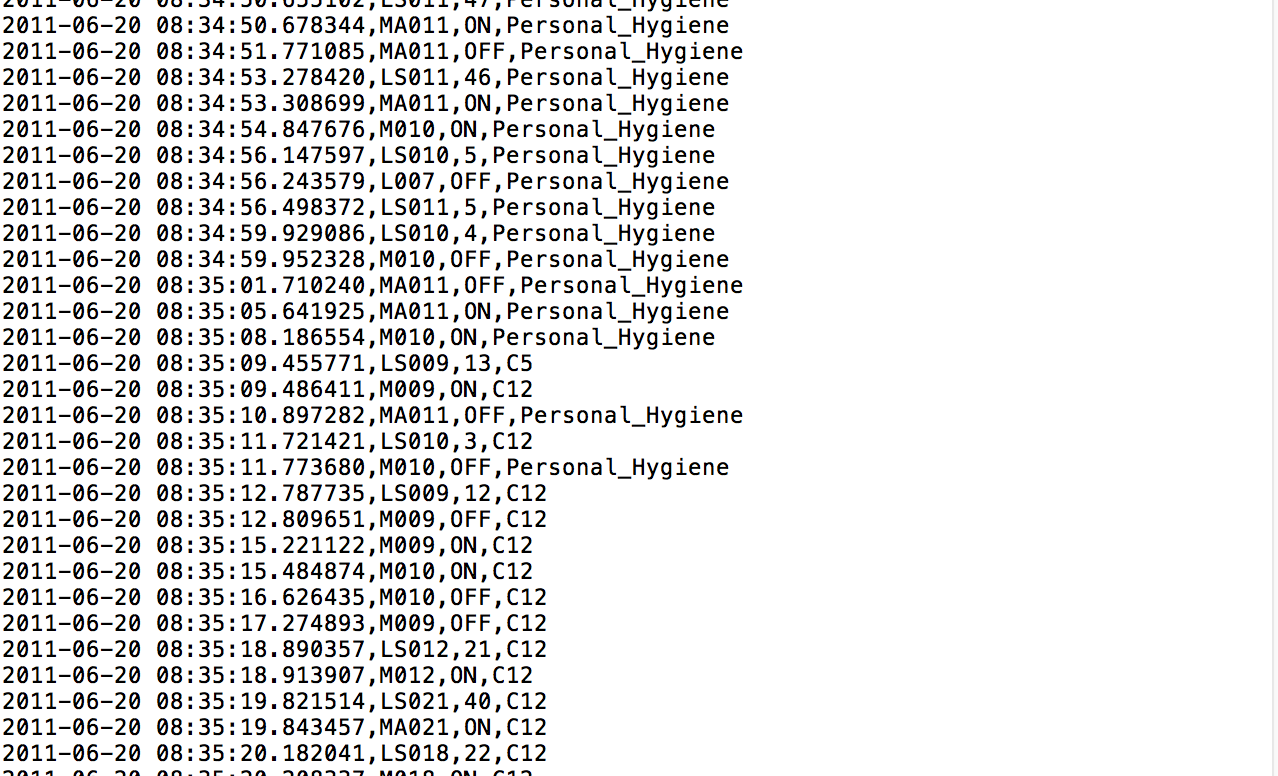
python runSynSys.py /path/to/originalsmarthomedata.csv <integer number of HMM states for activity HMM> <integer number of HMM states for sensor event HMM’s> <integer time interval in hours for timestamp reset> <integer to indicated type of model for timestamp generation>

See getTimestampModel() in SyntheticEventDataGenerator.py to see the models already there and feel free to add more.

example: python runSynSys.py /path/to/originalsmarthomedata.csv 10 10 6 1

**Input File format**

Currently, SynSys uses a csv file format for it’s input files like pictured below.



The file contains an ordered sequence of sensor events in the following format:

<date> <time> <sensor name> <sensor state or reading> <activity label>

If the real data file contains large amounts of Other Activity as a catch all for all unknown types of activities we recommend clustering these activities as a data preprocessing step.

**Output File format**

SynSys will print out synthetic data once the system has finished running in a csv file format similar to the original input file.